REMARKS

This Amendment is responsive to the Office Action dated April 7, 2006. Applicants have amended claims 1, 12, 23 and 30. Claims 1-30 are pending.

Claim Rejection Under 35 U.S.C. § 103

In the Final Office Action, the Examiner rejected claims 1-9, 12-15 and 18-30 under 35 U.S.C. 103(a) as being unpatentable over Jardin (USPN 6,681,327) in view of Friedman et al. (USPN 6,240,513). The Examiner also rejected claims 10, 11, 16, 17 and 29 under 35 U.S.C. 103(a) as being unpatentable over Jardin in view of Friedman et al. as applied to claim 1 above, and in further view of Abramson et al. (USPN 6,539,494).

Applicants respectfully traverse the rejection. The applied references fail to disclose or suggest the inventions defined by Applicants' claims, and provide no teaching or suggestion of Applicants' claims

Claims 1-4, 6-7, 9-13, 15-21, 23-30

In the previous response, Applicants explained a subtle but fundamental difference between Applicants' claim 1 and the cited prior art. Embodiments of Applicants' acceleration device incorporate a "direct mode" in which the acceleration device provides accelerated decrypting and forwarding of *application data* without requiring that the *application data* be reassembled at the application layer of the intermediate device. In other words, even though "application data" was encrypted at a client using conventional techniques, such as SSL, which operates at a layer above the packet level of the network stack, Applicant's intermediate device is nevertheless able to decrypt and forward the application data at the packet level. This "direct mode" is described, for example, with respect to FIG. 5, at pp. 14-18.

In the current Office Action, the Examiner suggested that Applicants' claims do not make it explicitly clear where in the network stack the client device encrypted the application data, and that the claims could be read so broadly that "application data" could be interpreted as application data that is merely transmitted through the application layer or the packet layer.[1]

Applicants have amended the independent claims to further clarify this distinction. In relevant part, amended claim 1 is directed to a method in which an intermediary device performs

[1] Office Action at pg. 2.

the steps of (d) receiving encrypted application data from a client via a secure communications session, *wherein the encrypted application data was encrypted by the client device by encrypting application data at a session layer above a packet level of a network stack of the client*; (e) decrypting encrypted application data; and (f) forwarding decrypted application data to the server via the secure network. Claim 1 also requires that the steps (e) and (f) are performed *at the packet level* of a network stack of the intermediate device *without processing the application data with an application layer of a network stack.* Thus, claim 1 requires that the intermediate device decrypt encrypted "application data" that was encrypted above a packet layer at the client, and forward the decrypted "application data" at the packet level without processing the application data with an application layer of a network stack.

As explained in the present application (see, e.g., FIG. 5, block 265), in a client, the application layer protocol hands unencrypted application data to the session layer. At the session layer, the client uses SSL to encrypt the *application data* and hands the encrypted application data down through the layers to the network IP layer, where it is finally divided into packets and introduced into the physical layers. As a result, a single SSL security record sent by the client may constitute multiple packets. Normally, these "multi-segment packets" of encrypted application data would need to be passed up above the packet layer of the network stack to form a single SSL record for reassembly before the application data can be decrypted. However, in direct mode, Applicants' acceleration device decrypts the application data at the packet level without requiring that the packets be passed up the network stack. Embodiments of Applicants' acceleration device have been modified to support a direct mode in which the TCP/IP layers (i.e., packet level) itself provides certain additional functionality to bypass the session-layer and application layer of the network stack and directly decrypt application data that was encrypted at higher levels of the stack.

As one example, pg. 10, ll. 10-15 describe certain features of the acceleration device that allow the upper levels of the network stack to be bypassed:

> *As shown in Figure 3, in accordance with the present invention, the SSL accelerator will intercept data destined for port 443 of the web server and, rather than the transmitting packets up and down the TCP/IP stack as shown in Figure 2B, will perform the SSL encryption and decryption at the packet level before forwarding the packet on to its destination.*

As another example, pg. 15, ll. 17-27 of the present application state:

> *During decryption, the device may utilize portions of its memory to buffer segments as necessary for decryption. The number and size of the buffers will depend on the cipher scheme used and the configuration of the packets, as well as whether the packets contain application data spanning multiple packets, referred to herein as multi-segment packets (and illustrated with respect to Figure 8). The SSL device can allocate SSL buffers as necessary for TCP segments. If, for example, application data having a length of 3000 bytes is transmitted via TCP segments having a length of 100 bytes, the device can, copy TCP segment 1 to a first SSL buffer, and start a timer, wait for packet 2 and when received, copy it to an SSL buffer and restart the timer, and finally when packet 3 is received, the SSL accelerator will copy it, decrypt all application data, authenticate it and forward the data on in the clear (emphasis added).*

In other words, Applicants' acceleration device can provide for full decryption and authentication of application data that was encrypted at the session layer or above without requiring the packets be collected and processed at the application layer of the network stack, thereby achieving certain performance advantages.

Jardin in view of Friedman fails to teach decrypting "application data" at a packet level without processing the application data at an application layer, as required by claim 1. The Examiner correctly recognizes that Jardin fails to provide any such teaching. Moreover, Jardin in view of Friedman does not teach or suggest techniques by which an intermediate acceleration device receives encrypted "application data" that was encrypted at a session layer *above a packet level*, and then decrypts that application data *at the packet level* without processing the application data with an application layer, as required by claim 1.

To the contrary, Friedman describes a network device where both encryption and decryption are applied to individual packets for purposes of network security, i.e., at the packet level. In other words, the security device of Friedman is not receiving encrypted "application data" (i.e., application-layer data that was encrypted) and then decrypting the "application data." Rather, Friedman is only encrypting and decrypting packet-layer data, i.e., data specific to individual packets.

For example, in the Summary, Friedman clearly states that "[i]t should be noted that encryption takes place at the IP level so that TCP and UDP packets are encoded" (emphasis added). Friedman further states that "[p]ackets received from the protected client are encrypted using an encipherment function such as IDEA, FEAL, or DES before being transmitted via the network to a destination" (emphasis added). Later in the Summary, Friedman describes how the

"data and proprietary tail" of a given packet are encrypted before transmission of that particular packet on the network. The IP level is well known to be network level (level 3) of the network stack, well below the application level (level 7) or the session level (level 5).[2]

Thus, it is clear that Friedman is not concerned with decryption and forwarding of encrypted application data, i.e., application-layer data that was encrypted at a higher level of the network stack above the packet level. Rather, both encryption and decryption are performed at the packet-level, i.e., the IP level. As a result, Friedman does not teach or suggest an intermediate acceleration device capable decrypting "application data" and forwarding the decrypted "application data" at the packet level without processing the application data with an application layer of a network stack.

Moreover, Friedman does not teach a solution to the problem addressed by Applicants' intermediate acceleration device, namely, an acceleration device capable of accelerating the processing of security records containing "application data." Friedmen fails to teach any mechanism capable of decrypting, at a packet level, security records of application data that was encrypted at a layer higher than the packet level. For at least these reasons, Jardin in view of Friedman fails to teach or suggest an intermediate device that receives "encrypted application data," i.e., application data that has originally been encrypted at the application layer, and decrypts and forwards the application data without processing the application data with an application layer of a network stack, as required by claim 1.

With respect to claim 12, for these or similar reasons, Jardin in view of Friedman fails to teach or suggest an apparatus in which a proxy SSL communications engine and a server TCP communications engine decrypt encrypted application data from the client at packet level within a network stack of the apparatus, and forward the decrypted application data to the one of the plurality of servers without processing the application data with an application layer of a network stack of the apparatus, wherein the encrypted application data was encrypted by the client at a layer above a packet level within a network stack of the client.

With respect to claim 23, Jardin in view of Friedman fails to teach or suggest receiving communications directed to the enterprise in secure protocol, wherein the secure protocol provides encrypted application data wherein the secure protocol provides encrypted application

[2] http://en.wikipedia.org/wiki/TCP/IP#The_layers

data that was encrypted by one of the customer devices at a session layer above a packet level within a network stack of the customer device, and then decrypting data packets of the secure protocol to provide decrypted packet data at the packet-level of a network stack of the intermediate device. Further, Jardin fails to teach or suggest bypassing the application layer of the network stack of the intermediate device and forwarding the decrypted packet data from the intermediate device to at least one server of the enterprise without processing the decrypted packet data with the application layer.

Similarly, with respect to claim 30, Jardin in view of Friedman fails to teach or suggest receiving encrypted application data that was encrypted by the client device by encrypting application data at a session layer or above within a network stack of the client, and then bypassing an application layer of a network stack of the intermediate device and forwarding decrypted application data from the intermediate device to the server via the secure network without processing the decrypted application data with the application layer.

For at least these reasons, the rejection of claims 1, 12, 23 and 30 under 35 U.S.C. 103(a) should be withdrawn. Claims 2-4, 6-7, 9-11, 13, 15-21, 24-29 are patentable for at least the reasons set forth above with respect to the independent claims on which they depend.

Claims 5 and 22

It appears the Examiner has not even considered Applicants' arguments previously presented on the record with respect to claim 5 and 22. Claims 5 and 22 require discarding at least a portion of each of the records after forwarding the portion to be discarded, and authenticating the decrypted application data of each data record using the remaining non-discarded portion of the data record upon receiving a final segment of the multi-segment record. Thus, claims 5 and 22 literally require dropping of some packets once forwarded and then subsequently authenticating the application data using only the remaining portion. As explained in the Applicants' application, this may be used in certain embodiments to reduce the buffering requirements of the intermediate device. See, for example, Applicants' specification at pg. 26, ln. 20 – pg. 27 at ln. 21 that describes a "bufferless" or "small buffer" approach.

In rejecting dependent claims 5 and 22, the Examiner asserted that Friedman teaches these elements. However, the cited portions of Friedman (col. 12, ll. 43-49, col. 13, ll. 12-21)

only describe the secure device dropping packets when the device has no key in its database to use to encrypt the packet or the key has expired. Friedman's suggestion of dropping packets when no valid decryption key is present, as relied upon by the Examiner, could simply not be used to authenticate an application-level security record after dropping some packets associated with that record. If no valid key is even present, thus causing the Friedman system to drop packets, how could a security record associated with those packets be validated at all? Moreover, given that Friedman is only concerned with authenticating individual packets, how can the Examiner rely on Friedman for teaching authenticating the decrypted application data of each data record using the remaining non-discarded portion of the data record upon receiving a final segment of the multi-segment record? Once a packet is dropped in Friedman, it can no longer be authenticated. Friedman clearly states that encryption / decryption and authentication occur on a per-packet basis. Thus, with respect to claims 5, 7 and 22, Jardin in view of Friedman does not teach or suggest authenticating decrypted *application data* using a non-discarded portion of a multi-segment security record for that authentication data.

Thus, contrary to the Examiner's assertion, Jardin in view of Friedman does not teach or suggest discarding at least a portion of each of the records after forwarding those portions to the server, and then authenticating the security record using only the non-discarded portion, as required by claims 5 and 22.

Finally, for motivation to modify Jardin in view of Friedman, the Examiner makes an unsupported statement that one of ordinary skill would have been motivated because discarding a portion of a record and authenticating the remaining portion will enable the system to identify and discard records that have been altered or modified without processing the complete record. The Examiner cites no evidence in the record that such a motivation is found in the prior art, as is required. Second, Applicants are at a loss as to the Examiner's apparent conclusion that Friedman's packet-level encryption and decryption technique would allow one to discard records without processing the complete record.

For at least these reasons, the rejection of Applicants' claims 5 and 22 under 35 U.S.C. 103(a) are clearly erroneous and should be withdrawn.

Claims 8 and 14

Applicants' claims 8 and 14 require prior to establishing a communications session with one of said plurality of servers, selecting one of said plurality of servers to forward the decrypted authentication data to based on a load balancing algorithm that calculates processing loads associated with each of the servers. Thus, Applicants' claimed acceleration devices controls the opening of new sessions based on currently calculated loads of the servers.

In contrast, Jardin describes a broker 130 that controls the flow of individual transactions to servers based on priorities associated with users. With respect to claims 8 and 14, the Examiner relied on column 8, lines 27-67 through col. 9, line 10 of Jardin. In the cited portions, however, Jardin first describes a "broker 130" capable of prioritizing transactions. Broker 130 monitors response times for individual transactions and then may elect to "reduce the flow of transactions" to one server over another. Controlling issuance of individual transactions is fundamentally different from actively load-balancing *sessions* across the servers by selecting the server based on current processing loads prior to establishing the communication session with the selected server, as required by Applicants' claims 8 and 14.

In rejecting these claims, the Examiner states that Jardin teaches different algorithms to balance loads. Although this may be true, the Jardin algorithms are directed to load balance individual "transactions" and not entire communication sessions.

CONCLUSION

All claims in this application are in condition for allowance. Applicants respectfully request reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date: May 22, 2006

By: Kent J. Sieffert

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

Name: Kent J. Sieffert
Reg. No.: 41,312